**Τίτλος:** «“Αλγόριθμοι και Πολυπλοκότητα”

Χρονοπρογραμματισμός εξετάσεων Πανεπιστημίου V1.0»

**Όνομα Καθηγητή:** Γκόγκος Χρήστος

**Τεχνική Αναφορά**

**Τίτλος Παραδοτέου:** Εργασία στο μάθημα “Αλγόριθμοι και Πολυπλοκότητα”

Χρονοπρογραμματισμός εξετάσεων Πανεπιστημίου V1.0

Συντάκτες:

1. Παγγές Ιωάννης Α.Μ:00089

Ημερομηνία: Ιανουάριος 2021

Πίνακας Περιεχομένων

[1.0 Περίληψη 3](#_Toc62129209)

[1.1 Εισαγωγή 4](#_Toc62129210)

[2.0 Χρονοπρογραμματισμός εξετάσεων Πανεπιστημίου 5](#_Toc62129211)

[2.1 Εκτέλεση προγράμματος 7](#_Toc62129212)

[3.0 Συμπεράσματα 8](#_Toc62129213)

[Βιβλιογραφία 9](#_Toc62129214)

[Παράρτημα Α 10](#_Toc62129215)

# 1.0 Περίληψη

Ο χρονοπρογραμματισμός των εξετάσεων είναι μια από τις πιο σημαντικές διοικητικές δραστηριότητες που πραγματοποιούνται σε όλα τα ακαδημαϊκά ιδρύματα. Η αποδοτική δημιουργία προγραμμάτων εξετάσεων είναι ένα σημαντικό και επαναλαμβανόμενο πρόβλημα το οποίο καλούνται να αντιμετωπίσουν τα εκπαιδευτικά ιδρύματα σε όλο τον κόσμο. Μια απλοποιημένη μορφή του προβλήματος έχει προταθεί από τους Carter κ.ά. οι οποίοι διέθεσαν δημόσια 13 στιγμιότυπα προβλημάτων που εν συνεχεία χρησιμοποιήθηκαν σε πληθώρα επιστημονικών εργασιών χρονοπρογραμματισμού. Στα πλαίσια του μεταπτυχιακού προγράμματος οδηγηθήκαμε να δημιουργήσουμε ένα πρόγραμμα για την λύση του χρονοπρογραμματισμού των εξετάσεων σε ακαδημαϊκά ιδρύματα.

**Λέξεις κλειδιά:** χρονοπρογραμματισμός, πρόγραμμα, εξετάσεις

# 1.1 Εισαγωγή

Το πρόβλημα αφορούσε κυρίως εξετάσεις, σπουδαστές και συνεχόμενες περιόδους σε κάθε μια από τις οποίες μπορούν να διεξαχθούν μια ή περισσότερες εξετάσεις. Κάθε εξέταση διαθέτει μια λίστα από σπουδαστές και κάθε σπουδαστής μπορεί να είναι εγγεγραμμένος σε μια ή περισσότερες εξετάσεις. Η λύση του προβλήματος συνίσταται στην ανάθεση εξετάσεων σε περιόδους έτσι ώστε να μην υπάρχουν συγκρούσεις, δηλαδή να μην υπάρχουν σπουδαστές που θα έπρεπε να συμμετάσχουν σε εξετάσεις σε περισσότερα του ενός μαθήματα στην ίδια περίοδο. Καθώς είναι ενδεχόμενο να υπάρχουν πολλά εναλλακτικά προγράμματα που ικανοποιούν τον ανωτέρω περιορισμό, προτιμότερο θεωρείται εκείνο το πρόγραμμα που διαθέτει επαρκή διαστήματα προετοιμασίας ανάμεσα σε διαδοχικές εξετάσεις για όλους τους φοιτητές συνολικά. Το πρόγραμμα υλοποιήθηκε κατά κύριο λόγο στη γλώσσα προγραμματισμού C.

# 2.0 Χρονοπρογραμματισμός εξετάσεων Πανεπιστημίου

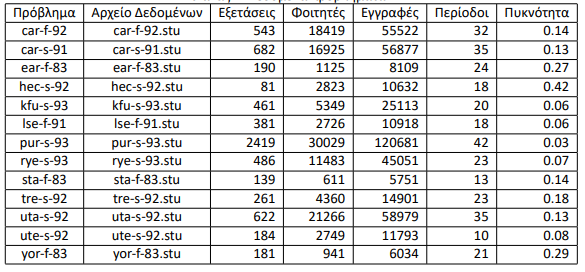
Τα προβλήματα χρονοπρογραμματισμού αποτελούν μια ιδιαίτερη κατηγορία προβλημάτων συνδυαστικής βελτιστοποίησης τα οποία εκδηλώνονται συχνά στην πράξη. Υπάρχουν πολλά προβλήματα χρονοπρογραμματισμού (χρονοπρογραμματισμός εκπαιδευτικών ιδρυμάτων, χρονοπρογραμματισμός προσωπικού ανθρώπινου δυναμικού, χρονοπρογραμματισμός αθλητικών γεγονότων).

Οι Burke, Kingston and De Werra(2004) έδωσαν ένα σημαντικό ορισμό για το πρόβλημα του χρονοπρογραμματισμού. Όπου ένας χρονοπρογραμματισμός προβλήματος αποτελείτε από τέσσερις βασικές παραμέτρους:

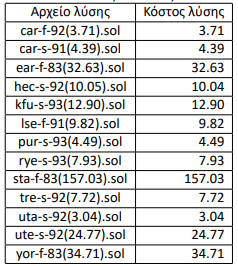
* **T** ένα πεπερασμένο σύνολο χρόνοθυρίδων (timeslot)
* **Ρ** ένα πεπερασμένο σύνολο πόρων (resource)
* **Μ** ένα πεπερασμένο σύνολο από εξετάσεις (meetings-exams)
* **C** ένα πεπερασμένο σύνολο από περιορισμούς (constraints)

Το πρόβλημα είναι πως μπορούμε να τοποθετήσουμε χρόνο και πόρους στο σύνολο των εξετάσεων έτσι ώστε να μπορέσουμε να ικανοποιήσουμε τους περισσότερους δυνατούς περιορισμούς, αυτό ονομάζεται το λεγόμενο NP πρόβλημα.

Στόχος της παρούσας εργασίας είναι να δημιουργήσουμε-αναπτύξουμε ενα πρόγραμμα που τα λύνει το πρόβλημα του χρονοπρογραμματισμού των εξετάσεων. Για να γίνει αυτό εφικτό πήραμε δεκατρία πραγματικά προβλήματα (University of Toronto Benchmark Data), έτσι ώστε να μπορέσουμε να δοκιμάσουμε τις λύσεις για να μπορέσουμε να τα συγκρίνουμε.



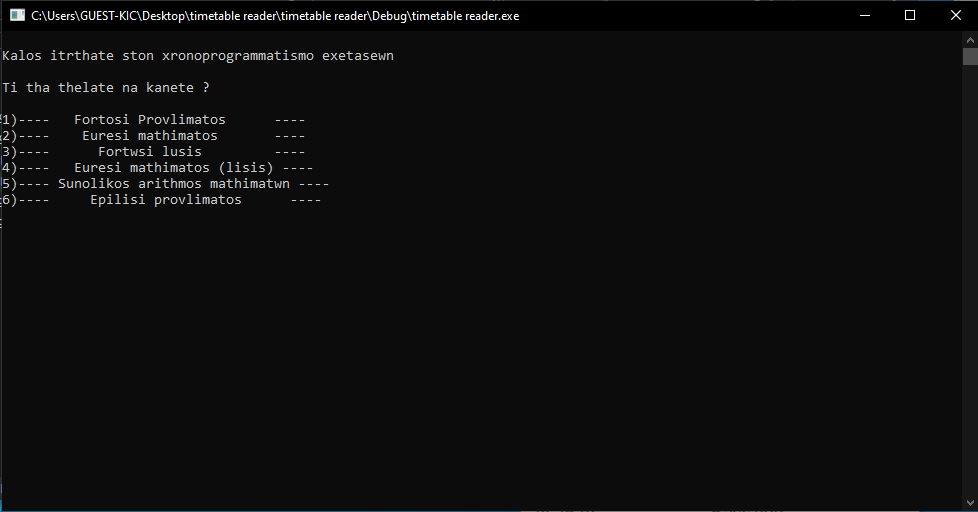
Εικόνα Δεδομένα Προβλημάτων



Εικόνα Κόστος Λύσεων

# 2.1 Εκτέλεση προγράμματος

Tα βασικά βήματα ώστε ο χρήστης να τρέξει ένα πρόβλημα του χρονοπρογραμματισμού των εξετάσεων θα περιγραφούν παρακάτω. Εκτελώντας το .exe μας παρουσιάζεται το shell. Η πρώτη επιλογή (Fortosi provlimatos) είναι για φόρτωση του αρχείου προβλήματος .stu που είναι τα μαθήματα. Η δεύτερη επιλογή (Euresi mathimatos), η οποία είναι χρήσιμη μόνο μετά την (1) είναι αναζήτηση του αριθμό των μαθητών που έχουν πάρει ένα μάθημα. Η επιλογή (3) είναι για φόρτωση του αρχείου .stu που είναι τα μαθήματα που έχουν πάρει όλοι οι φοιτητές του ιδρύματος. Μετά την εκτέλεση της υπάρχουν οι επιλογές (4-Euresi mathimatos lisis) και (5- Sunolikos arithmos mathimatwn) που είναι για αναζήτηση των μαθημάτων που έχει πάρει ένας τυχαίος φοιτητής καθώς και για την αναζήτηση του συνολικού αριθμού των μαθημάτων στο ίδρυμα. Έπειτα και αφού έχουν φορτωθεί τα δυο αρχεία του προβλήματος, με την επιλογή (6-Epilisi provlimatos) εισάγουμε αρχείο με κατάληξη .sol έτσι ώστε να συγκρίνουμε το κόστος λύσεων, με την επιλογή (6-Epilisi provlimatos) πρέπει να εισάγουμε και τον αριθμό περιόδου. Ο κώδικας της εφαρμογής βρίσκεται [εδώ](https://ioannispanges.github.io/xronoprogrammatismos/)



Εικόνα Γραφικό Περιβάλλον "Χρονοπρογραμματισμού Εξετάσεων"

# 3.0 Συμπεράσματα

Ο χρονοπρογραμματισμός των εξετάσεων είναι μια από τις πιο σημαντικές διοικητικές δραστηριότητες που πραγματοποιούνται σε όλα τα ακαδημαϊκά ιδρύματα. Πολλές μέθοδοι αναπτύχθηκαν, για την επίλυσή του, άλλες με επιτυχία και άλλες με όχι και τόσο καλά αποτελέσματα.

Η αντιμετώπιση του προβλήματος του χρονοπρογραμματισμού των εξετάσεων από τους εξελικτικούς αλγορίθμους, ήρθε με την παρατήρηση ότι αυτό το πρόβλημα μπορεί να παρομοιασθεί και να μετατραπεί σε πρόβλημα χρωματισμού ενός γράφου (graph colouring problem), κάτι το οποίο έχει επιτυχία. Τέλος, η εφαρμογή που αναπτύχθηκε εφαρμόστηκε σε μία σειρά πειραμάτων, πάνω σε δοκιμαστικά δεδομένα, και τα αποτελέσματα που προέκυψαν ήταν πάρα πολύ κοντά στις λύσεις.

# Βιβλιογραφία

[1] Κομηνός Χαράλαμπου-Γαβριήλ, “Υπολογιστικές εφαρμογές σε περιβάλλον παράλληλης επεξεργασίας,” Πανεπιστήμιο Πάτρας, 2013.

[2] Χ. Γ. ΓΚΟΓΚΟΣ, “ΑΛΓΟΡΙΘΜΟΙ ΣΥΝΔΥΑΣΤΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΜΕ ΕΜΦΑΣΗ ΣΕ ΜΕΤΑΕΥΡΕΤΙΚΕΣ ΤΕΧΝΙΚΕΣ- ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ,” ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ, ΠΑΤΡΑ, 2009.

[3] M. N. M. K. Ashis Kumar Mandal, *Solving Examination Timetabling Problem Using Partial Exam Assignment with Hill Climbing Search*. IEEE Xplore, 2015.

[4] T. U. of Nottigham, “Benchmark Exam Timetabling Datasets.” http://128.243.21.198/external/resources/index.html (accessed Jan. 20, 2021).

[5] The University of Nottigham, “Benchmark Exam Timetabling Datasets.” http://www.cs.nott.ac.uk/~pszrq/data.htm (accessed Jan. 20, 2021).

[6] R. Qu, E. K. Burke, B. McCollum, L. T. G. Merlot, and S. Y. Lee, “A survey of search methodologies and automated system development for examination timetabling,” *J. Sched.*, vol. 12, no. 1, pp. 55–89, Feb. 2009, doi: 10.1007/s10951-008-0077-5.

# Παράρτημα Α

* int choices(void): Συνάρτηση που δέχεται την αριθμητική επιλογή του χρήστη από το πληκτρολόγιο και την επιστρέφει σε ακέραιο.
* truct course \*readcourseslist(int \*globalcoursesaddress): Συνάρτηση που δέχεται σαν όρισμα την διεύθυνση μιας ακέραιης τιμής ώστε να αποθηκεύσει τον συνολικό αριθμό των μαθημάτων και επιστρέφει τον root Της Linked list με τα μαθήματα.
* struct student \*load\_student(void): Συνάρτηση που επιστρέφει τον root της linked list των μαθητών.
* void printcourses(struct course \*ptr,int crs\_to\_find): Συνάρτηση που δέχεται τον Root της linked list Των μαθημάτων και έναν ακέραιο Id προς αναζήτηση και αν υπάρχει στην Ram το αντίστοιχο μάθημα εκτυπώνει τα enrollments
* void printstudent(struct student \*ptr,int Idlookup): Συνάρτηση που δέχεται τον root Της linked list των μαθητών και το Id ενός μαθητή και αν αυτός υπάρχει εκτυπώνει τα μαθήματα που έχει πάρει αυτός.
* void printallstudents(struct student \*ptr): Συνάρτηση που παίρνει τον root της λίστας των μαθητών και τους εκτυπώνει όλους.
* struct courseintb\*\* Make2dintArray(int arraySizeX): Συνάρτηση η οποία δέχεται σαν όρισμα ένα μέγεθος και επιστρέφει έναν πίνακα δεικτών προς δομές τύπου Courseintb μεγέθους arraySizeX.
* void filltable(struct courseintb \*\*timetable,int\*\* conflictmatrix,int courses,int \*sortedmatrix,int timeslots): Συνάρτηση που έχει σαν σκοπό να γεμίσει το άδειο timetable ενός χρωμοσώματος. Δέχεται σαν ορίσματα το timetable, τον CD,τα μαθήματα, τον πίνακα του LD και τα timeslots
* double calculatecost(struct student \*ptr,struct courseintb \*\*timetable,int timeslots): Συνάρτηση υπολογισμού κόστους που επιστρέφει το κόστος και δέχεται σανορίσματα τον Root των μαθητών, ένα timetable για να υπολογίσει το κόστος του και τα timeslots.
* double costforstudent(int \*student,struct courseintb \*\*timetable,int timeslots): Συνάρτηση που υπολογίζει το κόστος για έναν μαθητή και δέχεται σαν ορίσματα τον πίνακα των μαθημάτων του μαθητή, το timetable και τα timeslots.
* int lookforcourse(int coursetofind,struct courseintb \*\*timetable,int timeslots): Συνάρτηση αναζήτηση μαθήματος(όρισμα 1) σε ένα timetable(όρισμα 2).Τελευταίο όρισμα είναι τα timeslots.
* void countcourses(struct courseintb \*\*timetable,int timeslots): Συνάρτηση που μετράει τα μαθήματα που βρίσκονται σε ένα timetable(1).
* double distancecost(int A,int B): Συνάρτηση που επιστρέφει το κόστος δυο μαθημάτων τύπου double που βρίσκονται στα timeslots Α,Β που είναι τα ορίσματα της.